TRAFFIC MONITORING WITH DIGITAL CAMERAS

A Thesis Submitted in Partial
Fulfillment of the Requirements
for the Degree of
DIIT

by

J Ravikantha Babu

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
KANPUR

April, 2002

TRAFFIC MONITORING WITH DIGITAL CAMERAS

A Thesis Submitted in Partial
Fulfillment of the Requirements
for the Degree of
DIIT

by

J Ravikantha Babu

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
KANPUR

April, 2002

Certificate

This is to certify that the thesis entitled '*Traffic monitoring with Digital Cameras*' by J Ravikantha Babu (Roll No Y022402) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree

[illegible]

K S VENKATESH
Asstt Professor
Deptt of Electrical Engineering
Indian Institute of Technology
Kanpur – 208 016

ABSTRACT

In this thesis work an image processing and object tracking approach is proposed for the analysis of a traffic monitoring system using pictures from digital cameras or videos Different approaches are explained for images obtained from a fixed camera The algorithms are developed to estimate traffic flow and speeds Traffic congestion has become a significant problem Early solutions attempted to lay more lanes to avoid congestion but adding more lanes is becoming less and less feasible

Contemporary solutions emphasize real time information and control to efficiently manage and use the existing infrastructure The quest for real time traffic information and thus an increasing reliance on traffic surveillance has resulted in a need for better vehicle detection Also most importantly this thesis work could lead to the better understanding and modeling of traffic flow since individual vehicle data (e g spacing velocity acceleration) can be observed In the first stage a three dimensional model of the background road image is generated In the tracking stage each vehicle in the scene is isolated and tracked over many frames and its data can be obtained Experimental results on frame sequences taken from road traffic will be presented

Acknowledgement

At the very first I am thankful to the Almighty God and my Lord Jesus Christ who provided me the understanding and wisdom to complete the course

It is a great pleasure to express my deepest gratitude to Dr K S Venkatesh my thesis supervisor for his invaluable guidance and constant support

I am also grateful to Prof Sumana Gupta for teaching me basics of Image Processing Moreover I express my gratitude to Prof C Dasgupta for encouraging me to learn Mat lab

Words are not enough to explain my feelings towards my parents and family for their love and inspiration Especially I am remembering the wholehearted support of my wife Supriya at a crucial time in our life

I am extremely grateful to the Directorate General of Doordarshan for giving me an opportunity to undergo this course at this prestigious Institution I also extend my sincere thanks to my batch mates lab mates and colleagues from Doordarshan

I am also thankful to the Doordarshan Kendra Lucknow for providing me all necessarv help in getting frame sequence of traffic video

Dedicated to

Doordarshan Kendra

CHENNAI

Contents

I INTRODUCTION

As we know a motion picture or television broadcast is a sequence of still frames that are displayed in rapid succession The frame rate necessary to achieve proper motion rendition is usually high enough to ensure a great deal of temporal redundancy among adjacent frames Much of the variation in intensity from one frame to the next is due to object motion The process of determining the movement of objects within a sequence of image frames is known as **motion estimation** Processing images accounting for the presence of motion is called motion compensated image processing

The problem of motion estimation from an image sequence has a variety of applications [3] In particular the estimation of multiple superimposed translational motions (displacements) has more recently received some attention Traffic monitoring meteorological monitoring of clouds and storms from satellite imagery and detection and tracking of airborne or ground based targets are all practical examples of the need for fast real time multiple motion estimation from video

In this project we have dealt with Traffic Monitoring Most approaches to this problem can be categorized as working either in the image domain (gradient based or region based) or in the spectral domain (Fourier transform based) Image (pixel domain) algorithms typically directed at short sequences of two or three images use the optical low brightness constraint (Optical algorithms used for regions of low brightness variations i e smooth regions) to estimate the motion parameters

Spectral approaches are based on the notion that if a sequence of images—thought of as a three dimensional (3 D) function on two dimensional (2 D) space and time contains a linearly moving pattern then the energy of the 3 D Fourier transform (FT) of this function will be concentrated along a plane through the origin whose orientation is related to the velocity of the moving pattern

Specifically in this work we present an image (pixel domain) algorithm that comprises

1) Processing the successive frames in the image sequence to get the background of image followed by

2) Computation for identifying the moving objects and tracking those objects in successive frames

3) An algorithm based on finding difference of position in two successive frames is applied to find out velocity and acceleration of moving objects and relating them to physical parameters

One of the applications of this work is image interpolation By estimating motion parameters we can create a new frame between two adjacent existing frames Another application is image restoration If we can estimate the motion parameters and identify regions in different frames where image intensities are expected to be the same or similar temporal filtering can be performed

This work will also become useful in video coding [2] It is well known that considerable temporal redundancy exists among consecutive video frames This temporal redundancy can be reduced by motion prediction and compensation techniques By predicting the current frame from the previous frames, we can limit our coding to the difference (image) in between the current frame and the predicted current frame The efficiency of this temporal redundancy removal depends on the efficiency and accuracy of the motion estimation In addition we may be able to discard some frames and reconstruct the discarded frames through interpolation from the coded frames

There are two classes of motion estimation methods block matching algorithms (BMA)[8] [9] and pixel recursive algorithms (PRA)[10] A PRA estimates motion on a pixel by pixel basis whereas a BMA estimates motion on a block by block basis Due to their implementation simplicity block matching algorithms have been widely adopted by various video coding standards such as CCITT H 261 [11] ITU T H 263 [12] and MPEG [13]

One straightforward BMA is the two dimensional (2 D) exhaustive search algorithm (ESA) The exhaustive search algorithm checks every possible motion vector candidate in a search window using a distortion measure and find the motion vector within that window that minimizes the distortion Although ESA finds the best motion vector in a global sense the large number of distortion calculations that it requires adds to the computational cost of video coders and limits the ESA s practical implementations Several fast BMA s have been developed with the intent of reducing the computation load including the three step search 2 D log search cross search one dimensional (1 D) full search and variations of the three step search These fast algorithms rely on the assumption that the motion compensated residual error surface is a convex function of the displacement motion vectors but this assumption is rarely true As a result these fast algorithms converge to a locally optimal displacement vector rather than the globally optimal displacement vector found by the ESA

In this work we have adopted PRA with some refinements to reduce its computational requirements Our basic approach is to analyze the sequence of frames from available video and try to estimate the velocity acceleration of each individual moving object in a busy traffic area and to provide useful information about collision avoidance and how to avoid traffic jams especially in our country s road conditions by knowing the traffic flow

II ALGORITHM-APPROACH

The motion estimation problem we consider here is of the translational motion of objects [1][4] Let $f(x\ y\ t_1)$ and $f(x\ y\ t_0)$ denote the image intensities at times t_1 and t_0 respectively We will refer to $f(x\ y\ t_1)$ and $f(x\ y\ t_0)$ as the past and current frame We assume that

$$f(x\ y\ t_0) = f(x\ dx\ \ y\ dy\ \ t_1) \qquad (1)$$

Where dx and dy are the horizontal and vertical displacement between t_1 and t_0

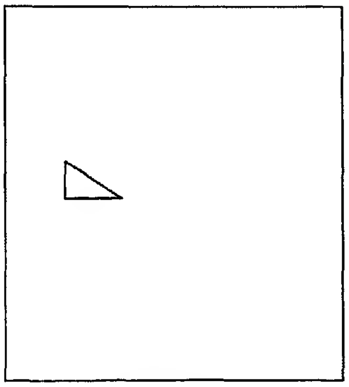


Fig $f(x\ y\ t_1)$

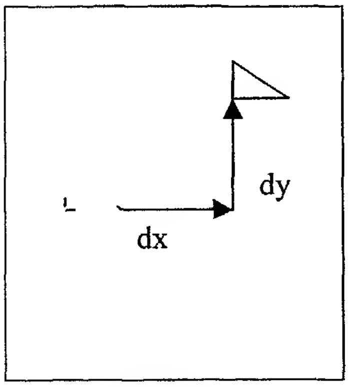


Fig $f(x\ y\ t_0)$

If we assume uniform motion then for any t in between t_1 and t_0

$$f(x\ y\ t) = f(x\ v_x(t\ t_1)\ y\ v_y(t\ t_1)\ t_1) \quad t_1 \le t \le t_0 \qquad (2)$$

Where v_x and v_y are the (supposedly) uniform horizontal and vertical velocities
A direct consequence of equation (2) is a differential equation which relates v_x and v_y to $\partial f(x\ y\ t)/\partial x$ $\partial f(x\ y\ t)/\partial y$ $\partial f(x\ y\ t)/\partial t$, which is valid in the spatio temporal region over which uniform translational motion is assumed

By deriving the relationship we land onto

$$v_x * \partial f(x\ y\ t)/\partial x + v_y * \partial f(x\ y\ t)/\partial y + \partial f(x\ y\ t)/\partial t = 0 \qquad (3)$$

Equation (3) is called a spatio temporal constraint equation The assumption of simple translation from equation (1) and the additional assumption of translation with uniform velocity that led to equation (3) are highly restrictive For example they do not allow for object orientation change camera zoom covering/uncovering of regions by translational object motion or multiple objects moving with different velocities

Motion estimation methods can be classified broadly into two groups that is region matching methods (based on equation (1)) and spatio temporal constraint methods (based on equation (3))

In this project we have adopted region matching methods

2 1 Region matching methods

Region matching methods [1] involve considering a small region in an image frame and searching for the displacement which produces the best match among possible regions in an adjacent frame In region matching methods the displacement vector (d_x d_y) is estimated by minimizing

$$\text{Error} = \int\int_{(x\ y\ \epsilon\ R)} | f(x\ y\ t_0) \quad f(x\ \ d_x\ \ y\ \ d_y\ t_1)|\ d_x\ d_y$$

with respect to d_x and d_y The function $f(x\ y\ t_0) - f(x\ \ d_x\ \ y\ d_y\ \ t_1)$ is called the displaced frame difference here R is the local spatial region used to estimate (d_x, d_y) The integrals in equation can be replaced by summation if f(x y t) is sampled at the integer spatial variables (x y)

Minimizing the error is a nonlinear problem Attempts to solve this nonlinear problem have produced many variations which can be grouped into block matching and pixel by pixel recursive methods Here we have adopted pixel by pixel recursive method for minimizing the error

2 2 Finding out the Background of an image sequence

Since we are mainly interested in the moving objects in a sequence the background in the images of the sequence only serves to confuse and interfere with our main objective the detection and tracking of moving objects The background consisting of some fixed objects like hoardings trees poles etc therefore is best deleted from the frames of the sequence

To evaluate the background we have taken a certain number of (4 or 5) consecutive frames and compared pixel values in each frame If the pixel value in one frame matches the corresponding pixel value in the majority of the other frames then we have retained that particular value (ie maximum occurrence) Taking into consideration some tolerance for these values we have obtained background from those particular values By taking more frames we can get even better results but at a correspondingly higher computational cost

The best way to get the background with the least computation is by making use of frames where the number of vehicles available on the road is less But such frames may not always be available

Here we have taken RGB space for analysis of each frame and by differencing background image from consecutive image frames we have got cleaned images And later on we have transformed each frame into YC_bC_r space which will be useful for getting brightness information

Our next step is to identify number of objects in each individual frame and estimate its motion parameters

2 3 Object identification

From each cleaned frame we have set the value of nonzero pixel values to 1 and the remaining to 0 (binarization) so that RGB values are not considered since we need only objects [6] Color is also important to identify which vehicle but information will be provided only on final result data

After binarization we have the images in black and white Now by applying the following technique we have separate out individual objects in each frame

$$\boxed{\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}}$$

Fig 1

First search for pixel value 1 in a frame And see if it has any adjacent 1 to it If it is so make them a group and repeat the process until you will land onto a zero in all directions Moreover we look only for groups having a considerable number of pixels so that it can make to a nontrivial object Groups that consist of just 2 or 3 pixels are rejected as they are more likely to have arisen out of minor perturbations in the background

Let X as cleaned frame and Y as the corresponding frames generated for each individual objects

> If X (i j) =1 and {X (i+1 j)=1 or X (i j+1)=1 or X(i+1 j+1)=1 or X(i 1,j)=1 or X(i j 1)=1 or X (i 1 j 1)=1 or X(i-1 j+1)=1 or X(i+1 j 1)}
>
> i e if X is having a nonzero neighbor
>
> Then Y (i j)=1

Next go to the neighbor pixel by incrementing or decrementing i j values i e position value where X(i j) is nonzero and find its corresponding nonzero neighbor whether it exists or not If it exists then take that value into Y(i,j) if it not exists then go to next nonzero neighbor and repeat the process until you will finish of collecting all the pixels We can understand this algorithm by looking into following examples

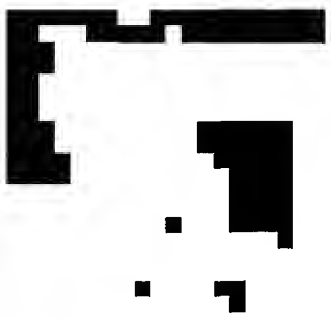
By applying the above algorithm to the above figure we ll obtain following figure

0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0
0	0	1	1	1	0	0	0
1	1	1	1	0	0	0	0
0	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0

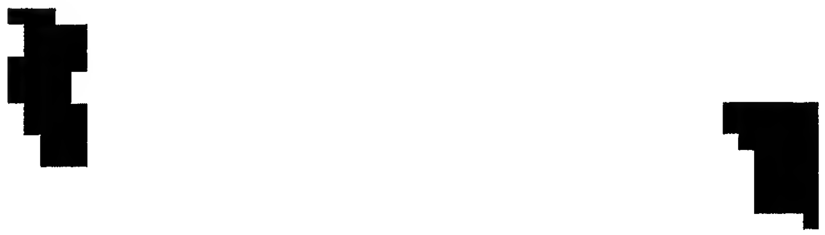
Fig 2

Moreover after getting first object from one particular frame to get the subsequent objects from the same frame again we have to follow the similar procedure But this time pixel value X(i j) is tested not only for non zero but also it should be tested whether it is already collected by first object or not If it is collected then it will skip to next pixel else it will create new object This procedure will be continued until all the objects were separated out from the existing frame By this method computational simplicity can be obtained By this method we have identified different objects in each frame By seeing the following illustration it will be clear

Original image



Images obtained after applying the algorithm mentioned



Now we made a list of all the available objects with a proper tag to each one of them for a particular frame We take the next frame and make a similar list by applying the above algorithm

Our next step ıs to ınterrelate the objects ın first frame lıst to that of the second one We have taken one object ın frame one lıst and compared ıt wıth the objects ın second lıst by applyıng followıng algorıthm

Let X ıs object 1 ın frame 1

For lıst 2 ın frame 2

If X(ı,j)=1

Is there a match for the same pıxel ın lıst 2?

And also does the sıze match to some extent to that of the object 1?

Then we can pıck thıs as the matched object for the object 1

Now streamlıne the second lıst correspondıng to that one of first lıst to the correspondıng matches Moreover there may be chances for the enterıng of new objects ın second frame Thıs ınformatıon would exıst ın the lıst 2 After matchıng each object ın lıst 1 ıf there stıll exısts some objects ın lıst 2 then we treat them as new entrants It may or may not be new entrant sınce ıf two objects whıch were mıxed (superposed) ın first frame can get separated ın second frame due to a velocıty dıfference among them and can appear as a new object Sımılarly we take frame 3 and frame 4 and we make matched lısts by correspondıng wıth the frame 1 lıst

Next we obtaın dıfference objects for each matched object wıth the same ıdentıficatıon tag as assıgned ın lıst 1 of frame 1

Dıfference ımage 1 = frame2 frame1

Dıfference ımage 2 = frame3 frame2

Dıfference ımage 3 = frame4 frame3

As we have stated above ınstead of subtractıng two cleaned frames for gettıng dıfference frame we have first separated ındıvıdual object from each cleaned frame and after obtaınıng these objects the followıng procedure followed Fırst for each object ın frame1 proper match was ıdentıfied ın frame2 lıst of objects and ıf match was found then we have subtracted the object (frame1) from matched object (frame2) ın thıs way dıfference objects were made for sımplıcıty though ıt ıs computatıonally costly

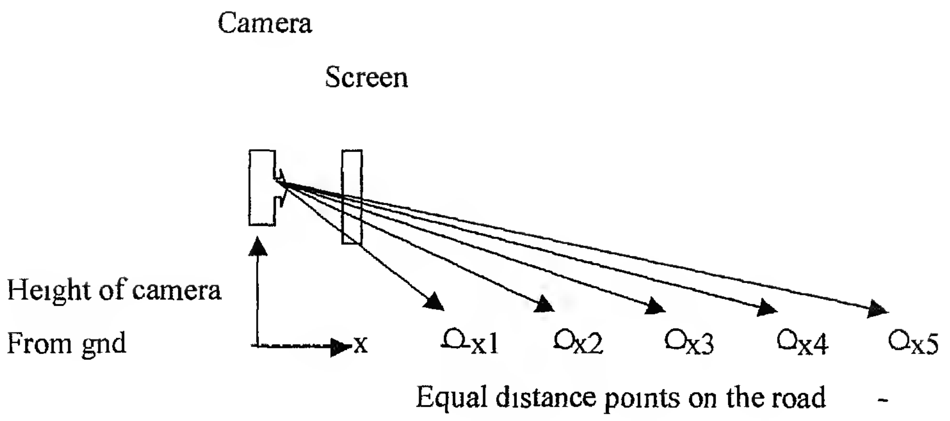
After obtaınıng three lısts of dıfference objects from four frames we estımate the sıze of these objects and centroıd and dısplacement and velocıty as explaıned below
For findıng out the sıze of an object we just add up all nonzero pıxels constıtutıng that object
To determıne the centroıd of an object we first sum up the entıre non zero x coordınates dıvıde by the sıze wıll gıve the x coordınate of centroıd Sımılarly we can get the y coordınate
And we make a correspondıng lıst for each object ıts sıze and ıts centroıd Thıs ınformatıon wıll say by how many pıxels and ın what dırectıon the object has been moved from one frame to the next frame

By applyıng geometrıcal rules we find the dıstance traveled by an object from one dıfference frame to the next dıfference frame And by dıvıdıng ıt by the frame perıod we obtaın the vırtual velocıty of an object on the screen

To get the physıcal velocıty of the objects the followıng algorıthm has been ımplemented

2 4 Physıcal velocıty of the objects

We assume that the camera heıght and angle towards the road are fixed and known Moreover we should also have the ınformatıon about length and breadth of the road that ıs vısıble to the camera ırrespectıve of road s ınclınatıon



x = distance from the lens to camera screen

By applying geometrical rules we make certain clusters in both horizontal and vertical directions according to the physical distance And we have obtained new coordinates in terms of physical distance for the each centroid of the difference frame objects Let us consider the above figure And the distances from camera to the respective distance points as x1 x5 (which are physically known) Now we made certain five clusters in each camera frame which will relate to the particular distance point

Say (in X direction) for i = 71 90 cluster 1—related to x5 x4 distance

Similar procedure for remaining clusters And this procedure will be followed for Y direction(j value) also with equal number of clusters

Now take the centroid coordinates of each object and check for both i,j values that in which cluster it will fall And we will assign corresponding physical distance value in X direction to x coordinate and distance value in Y direction to the y coordinate

Suppose i=75 then it falls in cluster 1 and its distance from camera is

x= x4+(x5 x4)/20*(75 71)

Similar procedure will be followed for getting y coordinate

Now we got the object coordinates in physical distance parameters from camera As per the above figure these clusters bunch in a close manner at a greater distance from the camera and separate out when close to the camera Thus even slow moving objects that are closer to the camera will have a greater apparent velocity and vice versa for objects distant from the camera So this position/velocity transformation map is made according to the camera viewing angle and height from ground It must be noted that this has to be done only one time and will apply as long as the camera position and orientation are not changed If the camera position is changed however we again have to determine the transformation map afresh It is also evident from the fact that the camera has a finite resolution that position/velocity measurements at points very far from the camera will become increasingly inaccurate

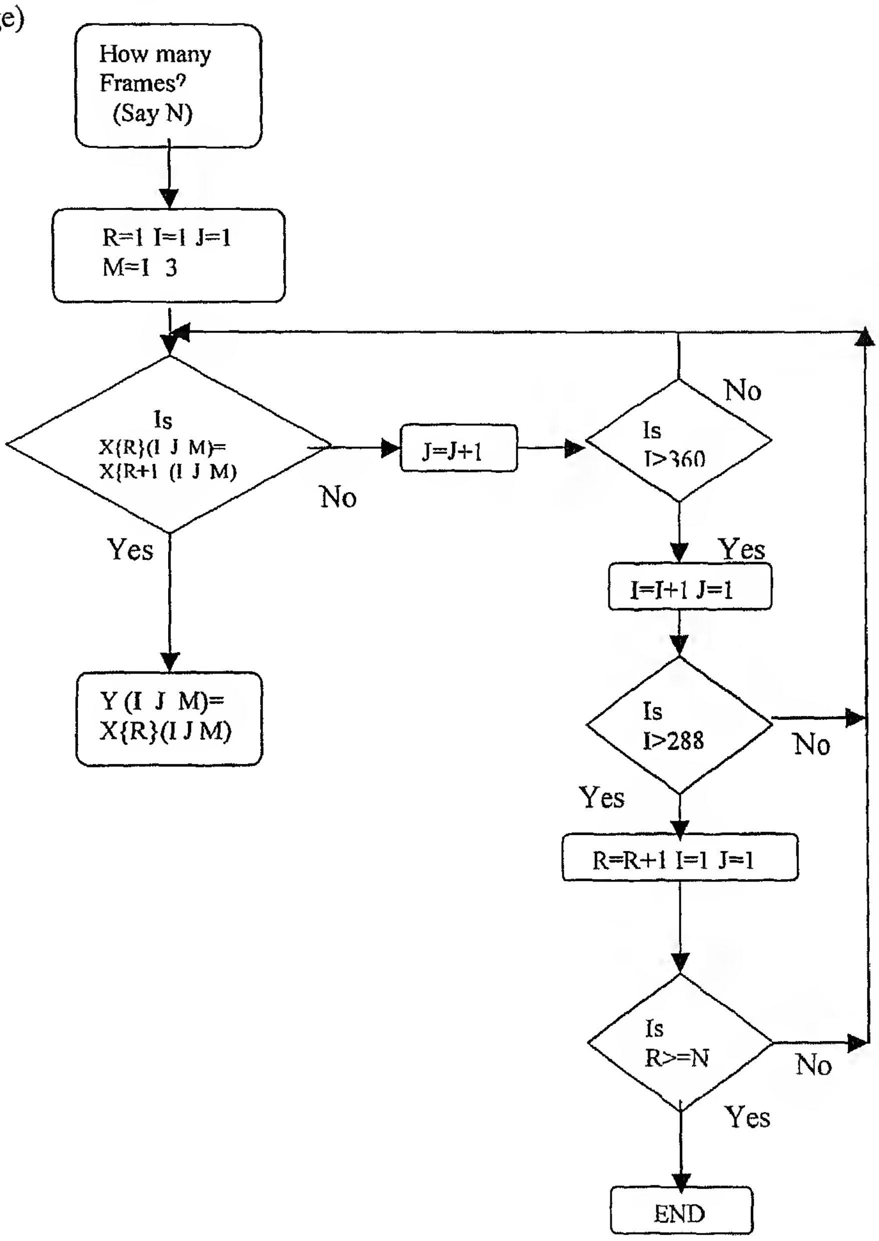
III IMPLEMENTATION AND RESULTS

3 1 Getting Background Image

Each frame is a 288x360 pixel 24 bit color image in tiff format The fixed camera is mounted on a pole or a tall building and overlooks a road or an intersection Since the camera is fixed the background image is also fixed We can assume we are given this background image that is also a 288X360 pixel 24 bit color image However even if we don t have this background image it can be easily generated A set of 20 40 initial frames may be taken The values of each pixel are monitored over the set of images The value that is most frequent for a pixel is typically its value for the background This is because as the vehicles and pedestrians move the background is revealed The objective is to detect the vehicle in consecutive frames and then to match the vehicles in consecutive frames so that they can be tracked Then knowing the distances traveled and the time intervals between the images we can find the speed of each vehicle
First we have taken some 20 frames of size 288 X 360 X 3 and obtained the background i e figure containing non moving objects like tree pole wall etc by using the algorithm discussed for generating background First frame pixel values have been compared with the consecutive frames and tolerance for these values has been considered up to +/ 6 And we have obtained the image containing background pixel values

By looking into the following flow chart we can understand the method

Flow Chart 1 (Note I=288 J=360 and M=1 3 for these frames ı e frame sıze X{R} R ınput frames and Y as output ımage)



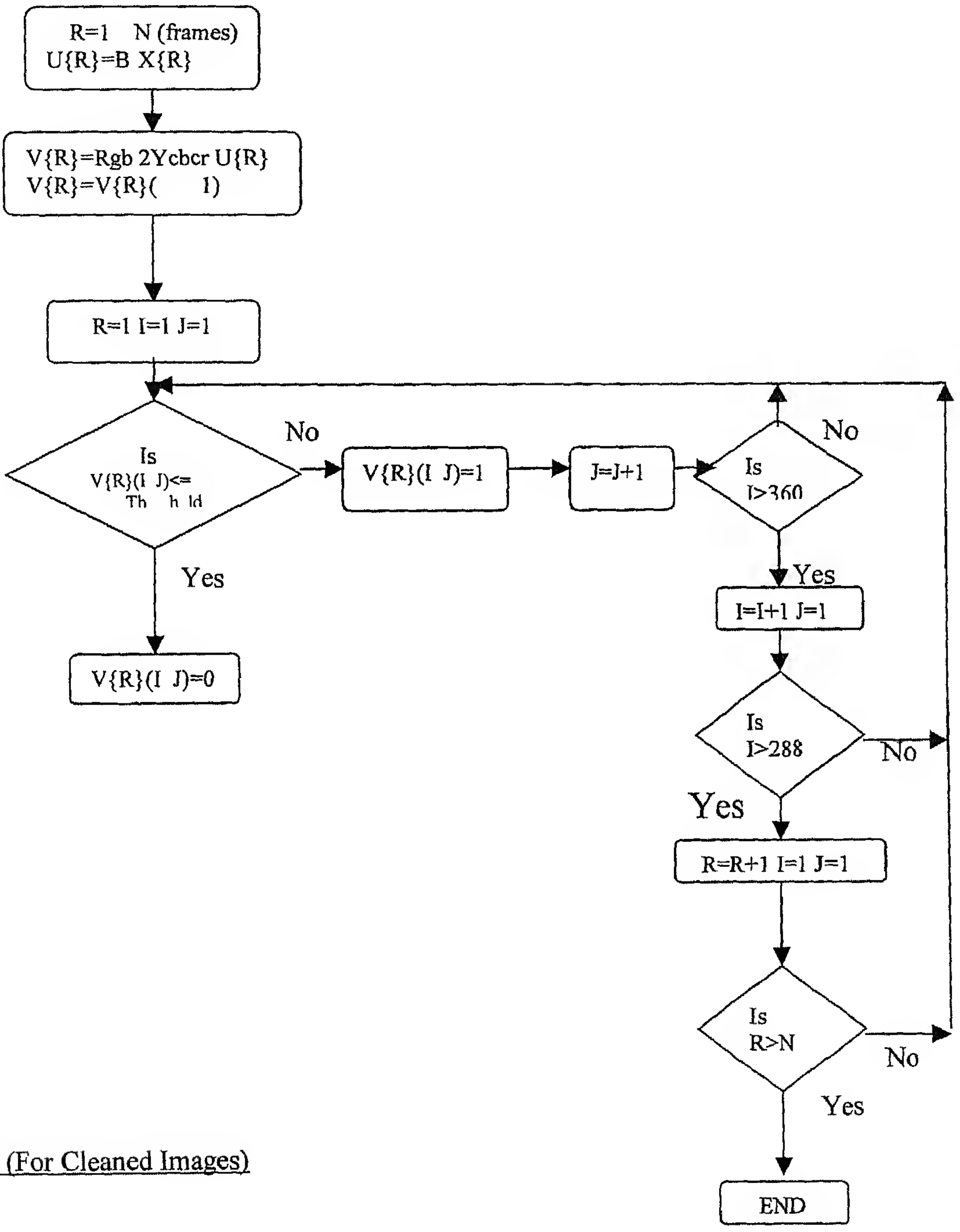
Flow Chart 1(For Gettıng Background Image)

3 2 Obtaining frames containing only moving objects (vehicles)

Each frame is subtracted from the above obtained background image and we have got frames having only moving objects Since these frames were in RGB space it is difficult to separate out each object on the basis of RGB values So for this YC_bC_r space was used and only Y space frames were considered for further processing After subtraction the frames ought to look black in the background areas (where moving objects are not present) but these black spaces are not exactly black but have certain nonzero values

By examining each frame a threshold pixel value (in this case we have chosen 25) was chosen and this value is common to all frames
Below this value the pixel value was made zero otherwise it was assigned to one
In this way we clean up the exiting frames and obtain pure black and white frames i e frames having pixel value either zero or one

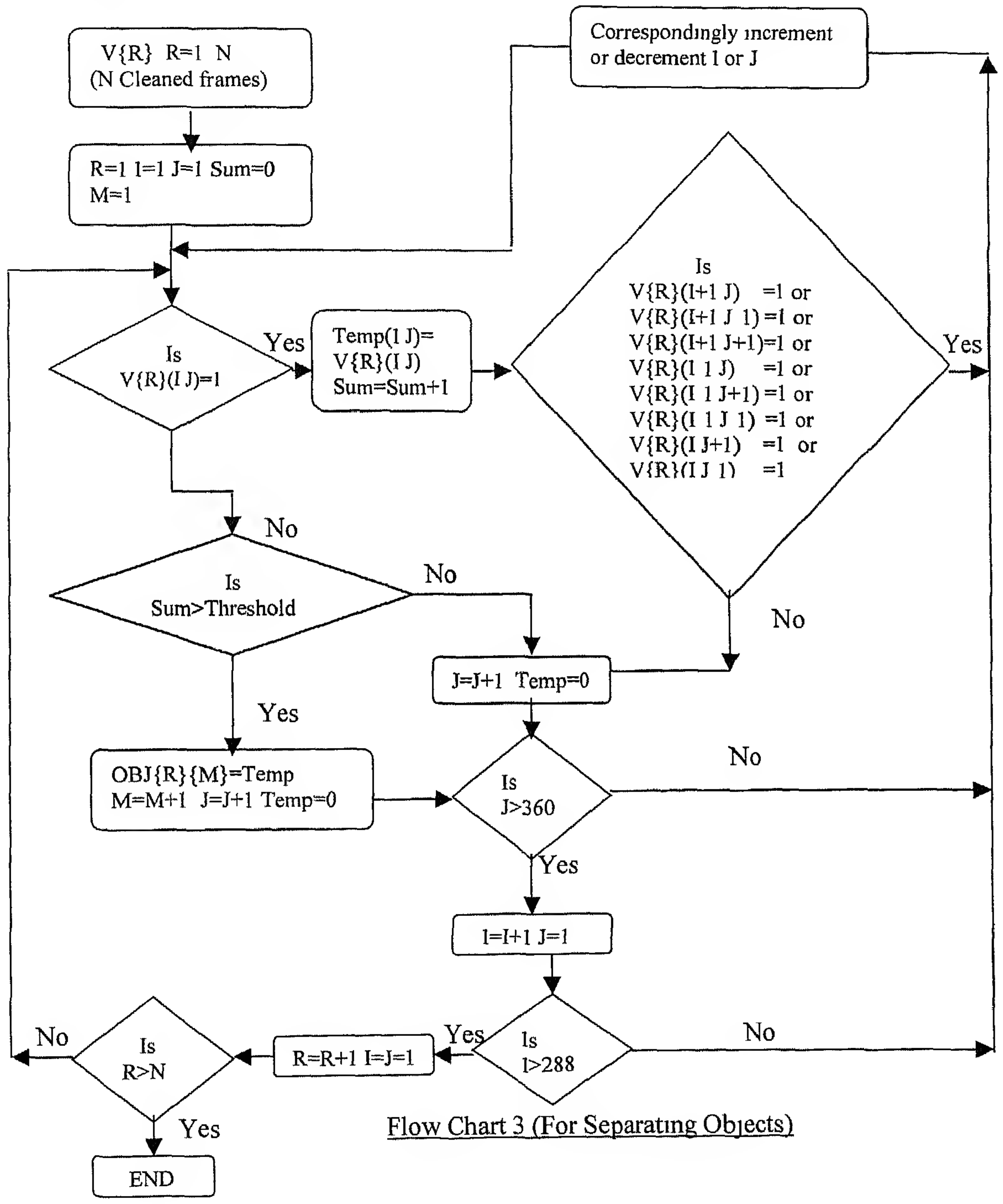
Flow Chart 2 (Note I=288 J=360 and for these frames ı e frame sıze X{R} R ınput frames and V{R} as cleaned output ımages B as Background ımage)



Flow Chart 2 (For Cleaned Images)

3 3 Separatıng out Objects

Now we have cleaned frames and by usıng the method mentıoned the objects were separated out from each frame

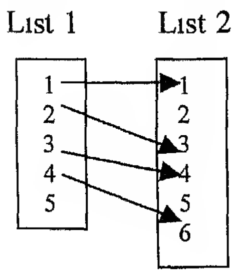


Flow Chart 3 (For Separatıng Objects)

Each frame was processed pixel by pixel and all the pixels having neighbor value one were collected There are eight neighbors for each pixel and all such non zero pixels were collected and tested for a minimum sized group of such pixels (Here we have chosen that threshold value as 100 which is common to all frames) below which it cannot be considered as object
Moreover the positions of the collected pixels are preserved and saved as separate images containing each separate object By this we have determined the number of objects present in a frame and assigned a tag to each object and prepared a list for all identified objects
A similar procedure is followed for the next consecutive frames and obtained objects with proper identity and we made list for each frame objects

3 4 Relationship Between Objects in Successive Frames

From the first frame the original pixel value of the first object is taken and compared with its counterparts pixel values in the objects of the next frame If any of the second frame objects pixel values matches that of the first object of the first frame then we have taken that one as the matched object By the same procedure we sort out all the objects in second list for corresponding objects in first list If no match is found then we link that particular object to 0 These are objects in the second list that do not have a counterpart in the first
Similarly by taking the sorted second list we have tested first for any missed objects that was not linked to the first list of objects and inserted those objects in rear end of the sorted second list
Example Let List-1(frame 1) has 5 objects and List 2 (frame 2) has 6 The corresponding matches were shown
Objects 1 1 2 3 3 4 4 6 was matched objects



After finding matched objects the two lists will be as shown below Especially List 2 will be modified as shown

List 1	List 2
1	1
2	3
3	4
4	6
5	0
	2
	5

Difference Obj 1= Obj1(list 1) – Obj1(list 2)

Difference Obj 2= Obj2(list 1) – Obj3(list 2)

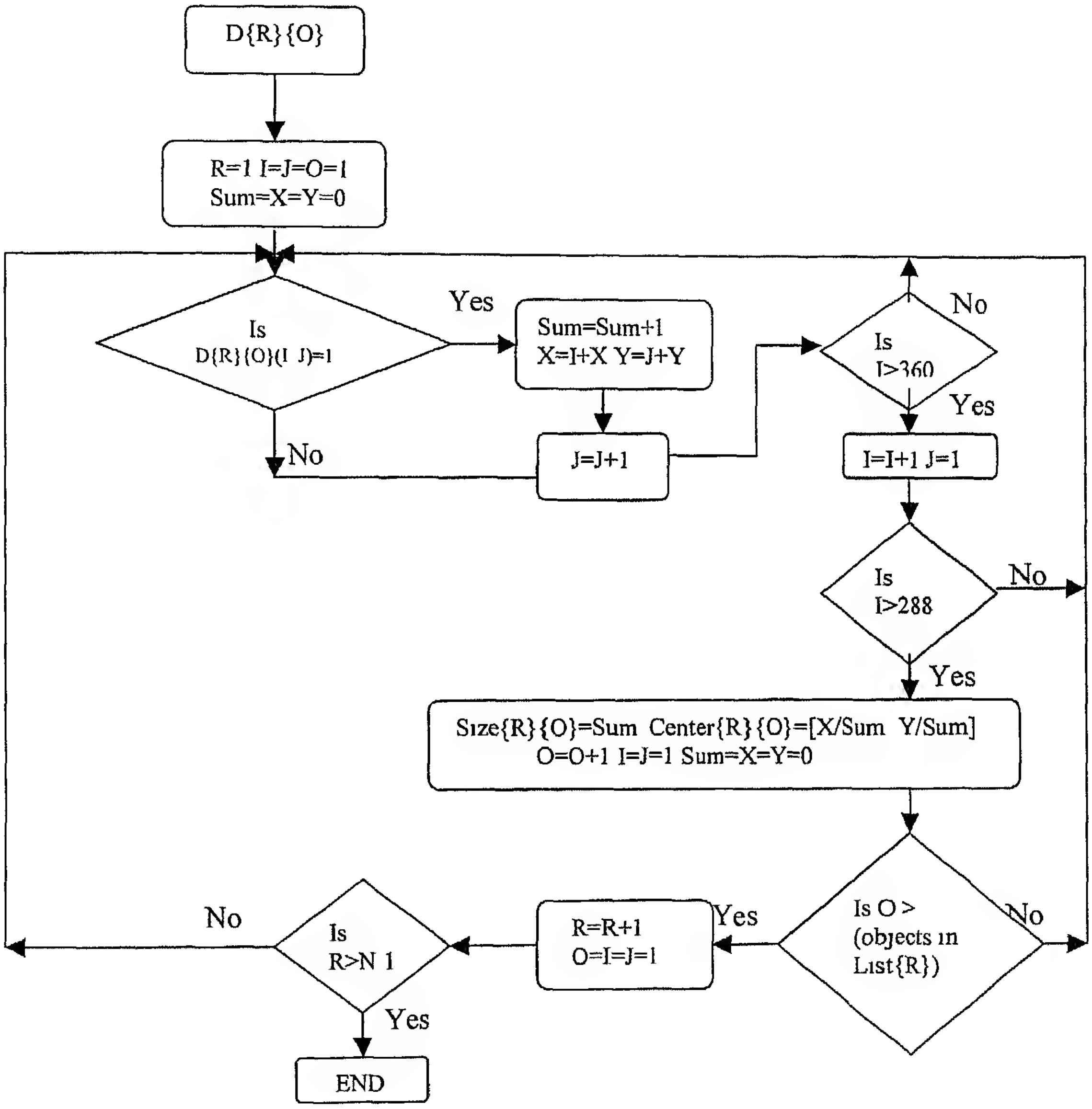
Difference Obj 3= Obj3(list 1) – Obj4(list 2)

Difference Obj 4= Obj4(list 1) – Obj6(list 2)

Difference Obj 5= No match found

Similarly sorted out list 2 objects will be taken and corresponding matched objects in list 3 will be found And this procedure will be continued for all frames If a match was found for objects then we have subtracted objects from each other and made difference objects were shown for above two lists These difference objects size and centroid were determined as per the flowchart mentioned First we have taken one object and summed up all non zero pixels in that object By this we can know total non zero pixels present in the object Later all the x positions of these non zero pixels were summed up And dividing this sum with total non zero pixels will give x coordinate of object Centroid Similar procedure followed for obtaining y coordinate By looking the following flow chart we can understand it clearly

Flow Chart 4 D{R}{O} difference objects R=1 N-1 (No Of frames) O= Number of difference objects in each list



Flow Chart 4 (For finding Size and Centroid of Objects)

3 5 Determining physical velocities

Now we have the size and centroid of each difference object These Centroid positions of each object are translated into physical parameters We have chosen six clusters for the existing camera position by treating horizontal viewing distance in camera as 30mtrs and vertical viewing distance in camera as 18mtrs (Approximately treating as supplied values) First we ve taken first object and took its x coordinate value of Centroid Then we ve checked for its cluster (in which cluster it will fall) and we ve assigned that particular physical position value (in meters) to the x coordinate Similarly we ve taken y coordinate value of the Centroid and obtained its physical position value Now we've translated all the Centroid values of each object into corresponding physical values

By using the following formula with frame rate we have obtained physical velocities of objects

Let x1= x position of difference object – x position of matched difference object

y1= y position of difference object – y position of matched difference object

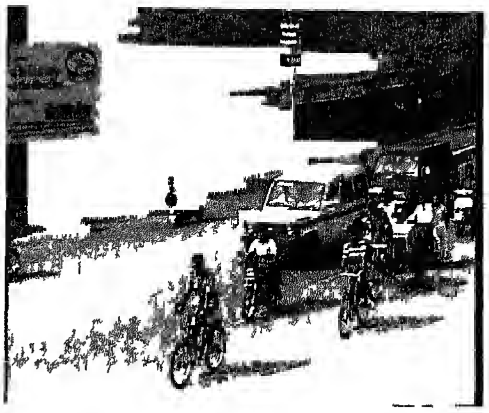
Displacement = $\sqrt{(x1*x1 + y1*y1)}$

Later on we have tested the algorithm for 20 successive frames and obtained the results and found them physically closer to real velocities

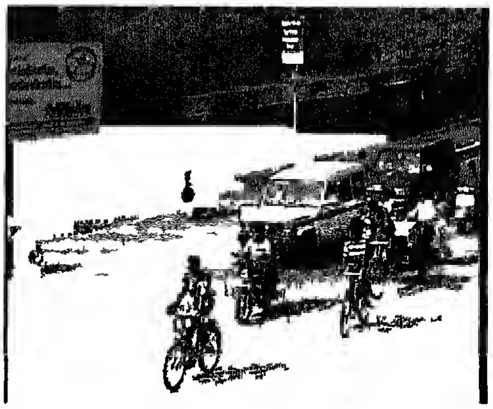
Some of the results with frames were presented here

The algorithms have been tested for 20 frames and obtained following results by running programs in MATLAB

frame 1


frame 2

frame 3


frame 4

frame 5

frame 6

frame 7

frame 8

Cleaned frames

frame 1

frame-2

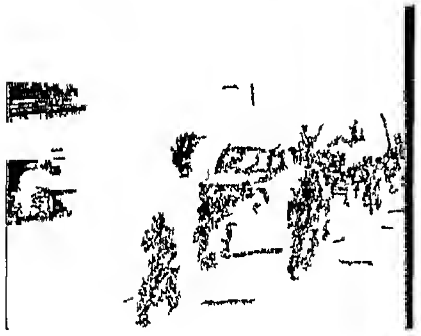
frame 3

frame-4

Cleaned frames

frame 5

frame-6


frame-7

frame-8

Separating out Objects from frame 1

Object 1

Object 2

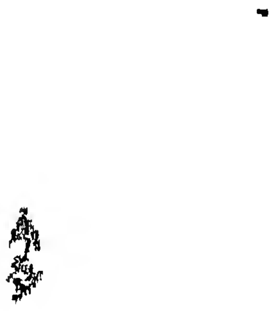
Object 3

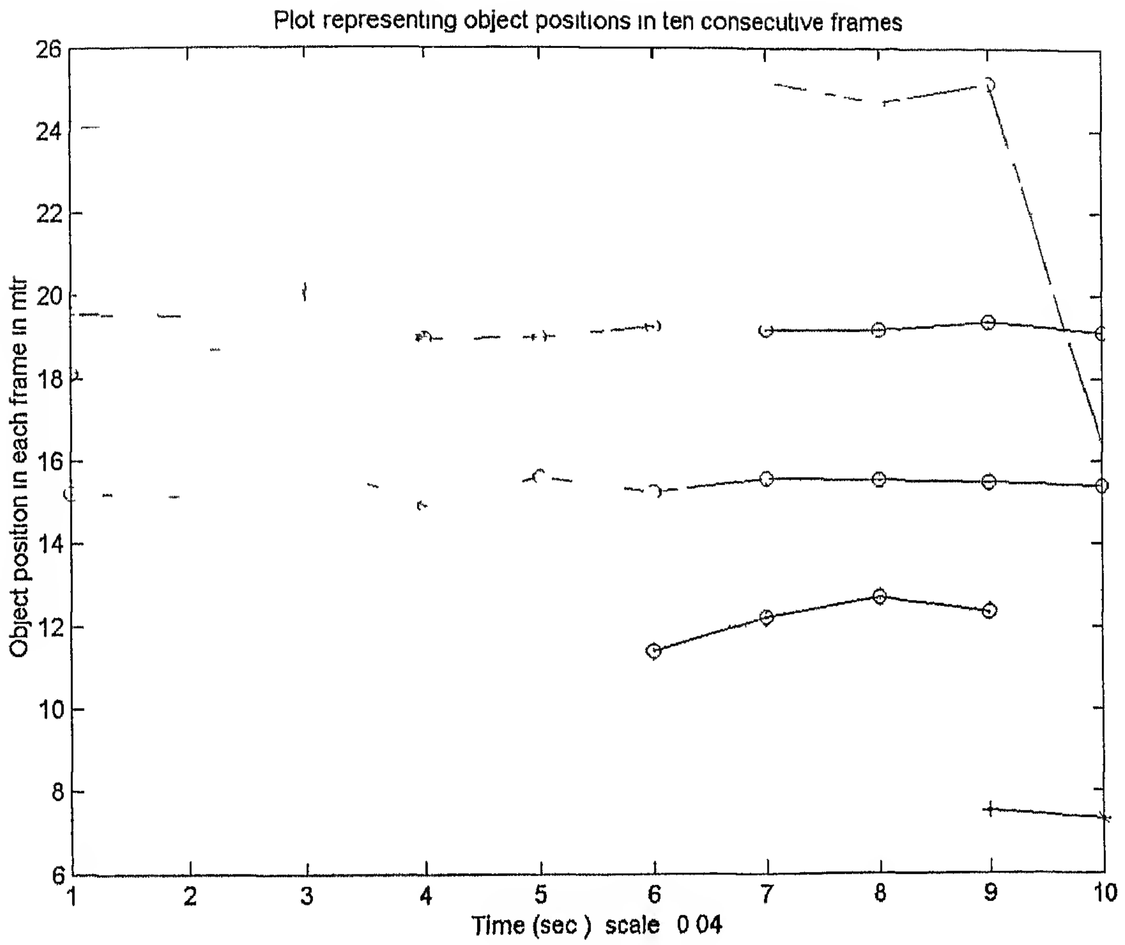
Object 4

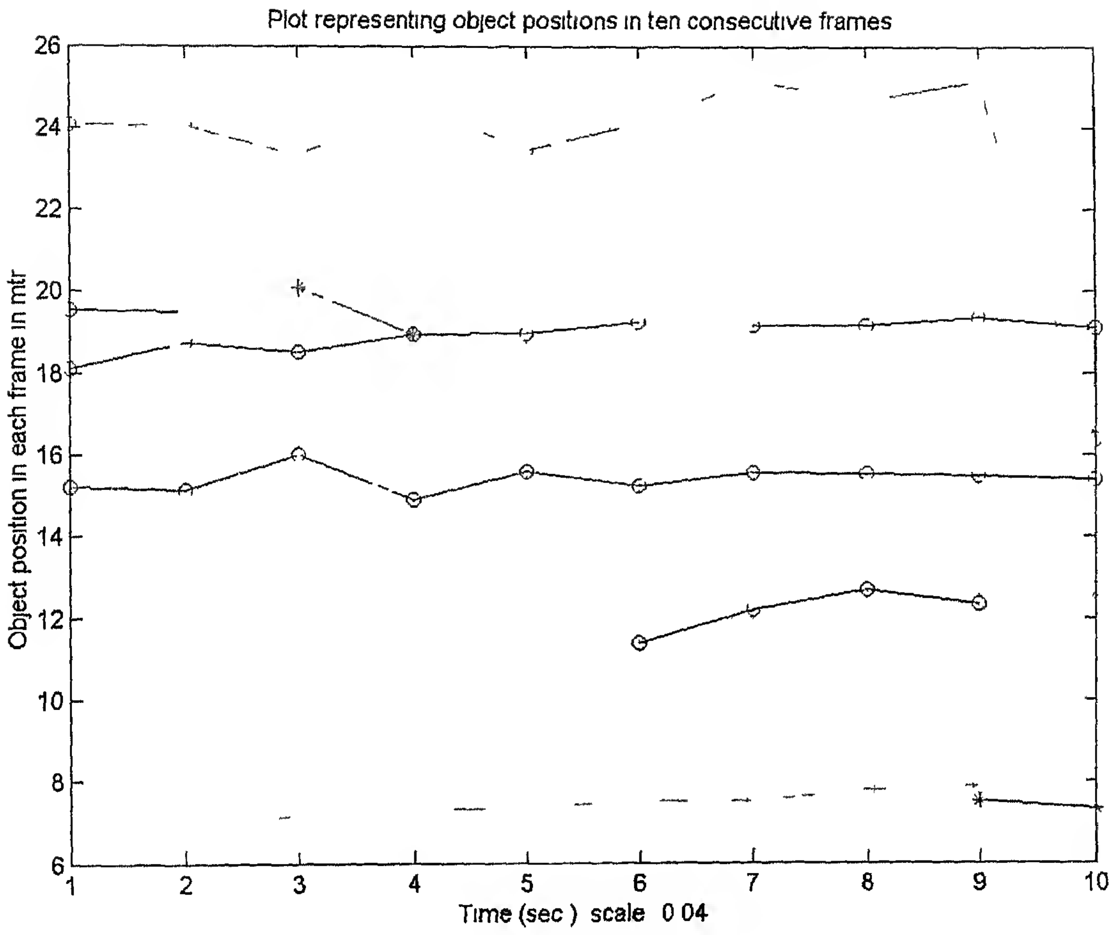
Separating out Objects from frame 1

Object 5

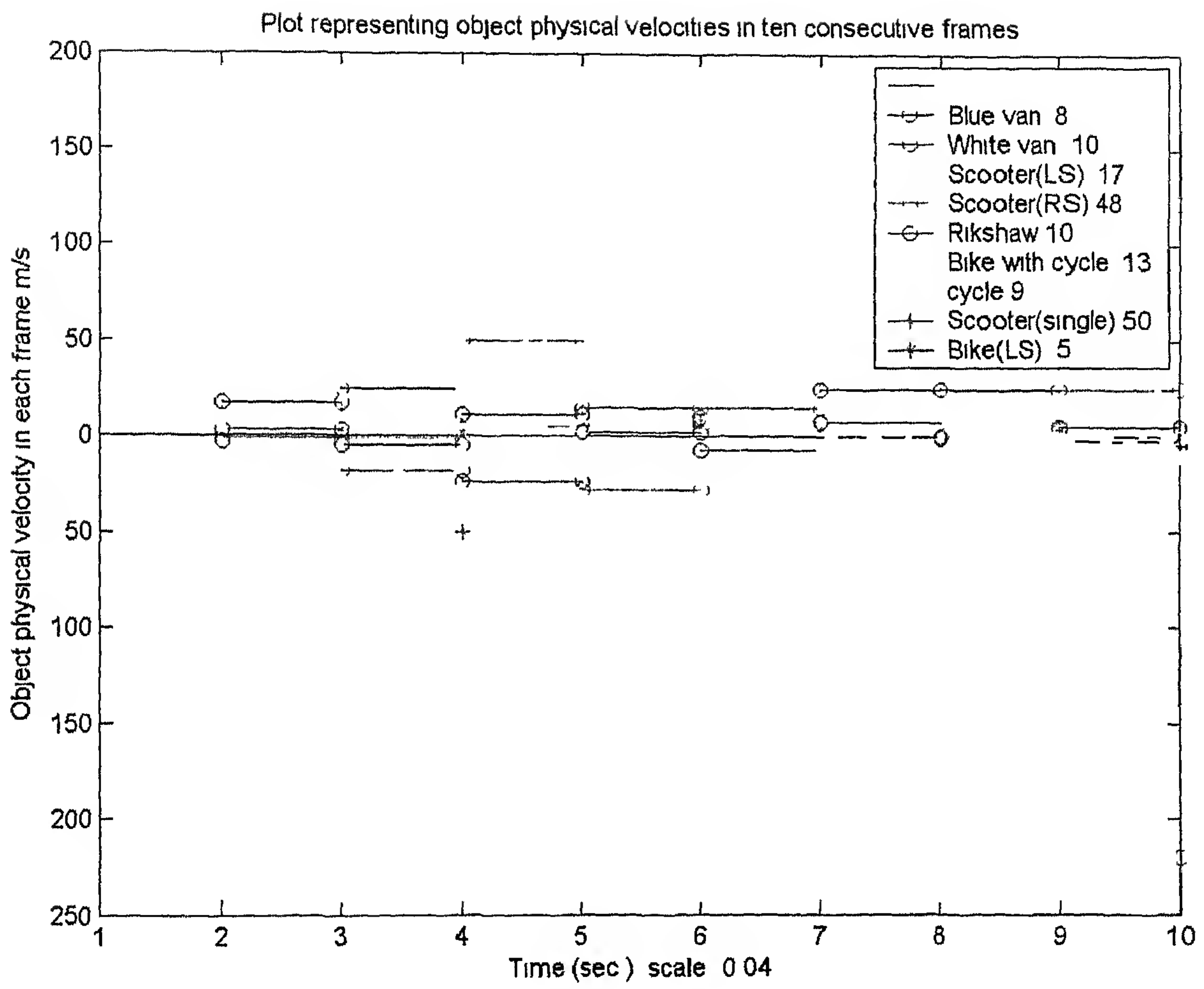
Object 6


Object 7


Plot representing object positions in ten consecutive frames
Object position in each frame in mtr
Time (sec) scale 0 04
26
24
22
20
18
16
14
12
10
8
6
1
2
3
4
5
6
7
8
9
10



Plot representing object positions in ten consecutive frames
Object position in each frame in mtr
Time (sec) scale 0 04
26
24
22
20
18
16
14
12
10
8
6
1
2
3
4
5
6
7
8
9
10


Plot representing object physical velocities in ten consecutive frames
Object physical velocity in each frame m/s
Time (sec) scale 0 04
200
150
100
50
0
50
100
150
200
250
1
2
3
4
5
6
7
8
9
10
Blue van 8
White van 10
Scooter(LS) 17
Scooter(RS) 48
Rikshaw 10
Bike with cycle 13
cycle 9
Scooter(single) 50
Bike(LS) 5

IV CONCLUSION & FUTURE WORK

Estimating velocities of objects finds vast application in traffic monitoring and management Especially in developed countries these cameras are all ready sending their video from different traffic points and this type of analysis will give a clear picture of law abiders and law breakers so that traffic jams collision avoidance and all sorts of streamlining can be done

Moreover as per the discussion it can also be helpful in finding out missing frames in between successive frames by image interpolation Also useful in image restoration and image coding

Only draw back is computational time since still it requires some fast processors so that it can be implemented in real time Moreover when two vehicles were closely related in one frame so that they will seem as one object when these will separate in next frame there are chances of error in deciding objects velocity This can be avoided by taking repetitive results

V REFERENCES

[1] J S Lim *Two Dimensional Signal Processing* Prentice Hall Englewood Cliffs N J 1990 Chapter 8 (Image enhancement)

[2] IEEE paper on *Fast algorithms for the estimation of motion vectors* By Huan sheng and R M Mersereau (IEEE Vol 8 March 1999)

[3] IEEE paper on *2 dimensional matched filtering for motion estimation* By Peyman Milanfar (IEEE Vol 8 March 1999)

[4] *Fundamentals of Digital Image processing'* by A K Jain Prentice Hall Englewood Cliffs NJ USA 1989

[5] Joe Palen (1997) *The Need for Surveillance in Intelligent Transportation systems* Caltrans New Technology and Research Program Vol 6 No 1

[6] *Real Time Estimation of Travel Speeds with Digital Cameras* Dinesh Chandnani Pitu Mirchandani Computer Science Dept ATLAS Research Center The University Of Arizona

[7] W Li and E Salari *Successive elimination algorithm for motion estimation* *IEEE Trans Image Processing* vol 4 pp 105–107 Jan 1995

[8] H G Musmann P Pirsh and H J Gilbert *Advances in picture coding* *Proc IEEE* vol 73 pp 523–548 Apr 1985

[9] J R Jain and A K Jain *Displacement measurement and its application in interframe image coding* *IEEE Trans Commun* vol COMM 29 pp 1799–1808 Dec 1981

[10] A N Netravali and J D Robbins *Motion compensated television coding Part I* *Bell Syst Tech J* vol 58 pp 631–670 Mar 1979

[11] CCITT Standard H 261 '*Video codec for audiovisual services at px64 kbit/s* ITU 1990

[12] ITU T DRAFT Standard H 263 *Video coding for narrow telecommunication channel at (below) 64 kbit/s* ITU Apr 1995

[13] ISO IEC JTC1/SC2/WG11 *Preliminary text for MPEG video coding standard* ISO Aug 1990